Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Уральский государственный педагогический университет»

Институт математики, физики, информатики и технологий

Кафедра информатики, информационных технологий и методики обучения информатике

ОБУЧЕНИЕ ПРОГРАММИРОВАНИЮ С ИСПОЛЬЗОВАНИЕМ ВИЗУАЛЬНОЙ СОБЫТИЙНО-ОРИЕНТИРОВАННОЙ СРЕДЫ SCRATCH В КУРСЕ ИНФОРМАТИКИ И ИКТ

Выпускная квалификационная работа
бакалавра по направлению подготовки
44.03.01 – Педагогическое образование

Профиль «Информатика»

Квалификационная работа
допущена к защите
Зав. кафедрой
_____ _____
дата подпись

Исполнитель: Пономарев М.В. ИНФ-1501
ИМФИиТ
_____
подпись

Руководитель: к.п.н., доцент кафедры
ИИТиМОИ Рожина И. В.
_____
подпись

Екатеринбург 2019

Оглавление

Введение

Базовый курс информатики начинается с 7 класса. Начинаются занятия программированием в рамках содержательной линии «Алгоритмизация и программирование» с освоения языка Pascal и среды программирования PascalABC.NET. Однако выбор среды программирования не так очевиден при проведении уроков по программированию на более ранней ступени обучения, если мы работаем с учениками в рамках непрерывного курса информатики, начиная с 5 класса [22].

При обучении программированию, может встать вопрос о выборе средств для обучения программированию, одним из которых является среда программирования.

Наряду с известными средами, такие как ЛогоМиры или КуМир, которые могут являться ограниченными в рамках своих задач, есть другие среды, не так часто упоминаемые или используемые, которые, однако предоставляют много возможностей для программирования и творчества и вместе с тем активно развиваются. Одной из таких сред служит Scratch.

Выбранная тема является в настоящее время актуальной, так как:

1. Обучение программированию способствует развитию логического и алгоритмического мышления и развитые логическое и алгоритмическое мышление является одним из требуемых результатов освоения курса информатики по ФГОС.
2. В сравнении со средами, которые рекомендуются к использованию в рамках пропедевтического курса, в частности ЛогоМиры и КуМир - Среда программирования Scratch является более наглядной, яркой и - что наиболее важно – активно развивающейся.
3. Среда программирования Scratch обладает большим набором инструментов: средствами программирования, графическим редактором, редактором звуков, средствами взаимодействия, что

позволяет разнообразить процесс выполнения учебных заданий, развивать творческие способности.

Тема: обучение программированию с использованием визуальной событийно-ориентированной среды Scratch в курсе информатики и информационно-коммуникационных технологий.

Объект исследования: процесс обучения информатике

Предмет исследования: обучение программированию в визуальной событийно-ориентированной среде Scratch

Цель: разработка учебных проектов для обучения в среде Scratch

Задачи:

1. Анализ литературы с целью выявления психолого-педагогических особенностей учащихся 5-6 классов;
2. Анализ известных сред обучения программированию применяемых в курсе информатики и ИКТ;
3. Изучение визуальной событийно-ориентированной среды программирования Scratch;
4. Разработка проектов и методических рекомендаций для учителей по применению разработанных материалов в обучении;
5. Апробация разработанных материалов

Глава 1. Теоретические основы обучения программированию в среде Scratch

1.1 Психолого-педагогические особенности учащихся

Согласно ФГОС среднего общего образования одним из результатов освоения информатики является сформированность основ логического и алгоритмического мышления [23]. Мышление вообще, это психологический процесс познания, связанный с открытием субъективного нового знания, с решением задач, с творческим преобразованием действительности. [15]

Под логическим мышлением мы будем понимать процесс, в ходе которого человек прибегает к логическим понятиям, основанным на доказательности и рассудительности. Его цель – получить обоснованный вывод, исходя из конкретных предпосылок [10]. Признаком развитого логического мышления может служить уверенное владение следующими приемами мыслительной деятельности:

- дедукция – выведение следствия из одного или нескольких утверждений, переход от общего к частному;
- индукция – выведение следствия из частных утверждений общего умозаключения, переход от частного к общему;
- анализ – процесс мысленного разделения объекта на части;
- синтез – процесс воссоединения целого из частей;
- сравнение – установление сходства и различия предметов и явлений;
- аналогия – перенос знаний об одном объекте, на другой объект;
- абстрагирование – процесс мысленного отвлечения от некоторых признаков, сторон конкретного с целью лучшего познания его;
- конкретизация – процесс возвращения мысли от общего и абстрактного к конкретному с целью раскрытия содержания;
- обобщение – определение общего понятия, в котором находит отражение главное, основное [17].

Мыслительная деятельность всегда направлена на получение какого-либо результата. Человек анализирует предметы, сравнивает их, абстрагирует отдельные свойства с тем, чтобы выявить общее в них, чтобы раскрыть закономерности, управляющие их развитием, чтобы овладеть ими. Другими словами, для того, чтобы получить истинное знание о некоем предмете, необходимо правильно использовать вышеперечисленные приемы, а значит иметь развитое логическое мышление.

А.И. Газейкина выделает алгоритмический стиль мышления, говоря о том, что это специфический стиль мышления предполагающий умение создать алгоритм, для чего необходимо наличие мыслительных схем, которые способствуют видению проблемы в целом, ее решению крупными блоками с последующей детализацией и осознанным закреплением процесса получения конечного результата в языковых формах [20].

Поняв, что такое логическое и алгоритмическое мышление, рассмотрим особенности развития в подростковом возрасте и как влияет программирование на развитие логического и алгоритмического мышления.

И.В. Шаповаленко говорит о том, что младший подростковый возраст характеризуется возрастанием познавательной активности, расширением познавательных интересов и «пик любознательности» приходится на 11 – 12 лет, что соответствует 5 – 6 классу. [24]

Теоретически, в рамках системно-функционального подхода, в отечественной психологии считается, что в подростковом возрасте ведущей функцией является развитие мышления. Под влиянием обучения, усвоения более обобщенных знаний и основ наук высшие психические функции постепенно преобразуются в хорошо организованные, произвольно управляемые процессы.

На данном этапе развития, у ребенка качественно улучшаются такие параметры внимания, как: объем, устойчивость, интенсивность, возможность распределения и переключения. Память внутренне опосредствована

логическими операциями; запоминание и воспроизведение приобретают смысловой характер. Увеличивается объем памяти, избирательность и точность мнемической деятельности. Постепенно перестраиваются процессы мышления – оперирование конкретными представлениями сменяется теоретическим мышлением. Данные новообразования должны появляться у подростка в идеальной ситуации, когда подросток развивается в благоприятных условиях, что происходит далеко не всегда.

Комплексное исследование, проведенное в средних и старших классах школы, показало, что достижения многих школьников весьма далеки от теоретически возможных.

Реальные семиклассники имеют невысокий уровень общего психического развития. Познавательная потребность у них бедна и однообразна, преобладают занимательные и пассивные формы ее удовлетворения. Общекультурные интересы достаточно широки и неустойчивы. Школьники опираются на способы механического запоминания, недостаточно используя приемы смыслового запоминания. Они не владеют в достаточной мере интеллектуальными приемами и умениями (вербального анализа, обобщения, образного анализа и синтеза). Теоретическое понятийное мышление развито слабо.

Поэтому одним из важнейших значений для развития мышления имеет организация и мотивация учебной деятельности в средних классах и содержание учебных программ.

По словам Д.Б. Эльконина, ведущей деятельностью данного периода развития является общение со сверстниками. В период младшего подросткового возраста деятельность общения выделяется в относительно самостоятельную область жизни. Главной потребностью этого периода является нахождением своего места в обществе, и данная потребность реализуется в сообществе сверстников. [1]

В отрочестве, как хорошо известно, общение со сверстниками приобретает совершенно исключительную значимость. В отношениях исходного возрастного равенства подростки отрабатывают способы взаимоотношений, проходят особую школу социальных отношений. [14]

В 10-11 лет, или в 5 классе, желание общения со сверстниками выражается в желании быть в среде сверстников, что-то делать вместе. Также в это время им важно получить у других людей оценку своих возможностей. Поэтому они стремятся заниматься той деятельностью, которая похожа на деятельность взрослых или найти искать тот вид деятельности, который имеет реальную пользу и получающий общественную оценку. [24]

Большим подспорьем в повышении мотивации учащихся в учебной деятельности может служить организация творческой деятельности учащихся, но прежде чем говорить об этом, разберем уровень развития творческих способностей в младшем подростковом возрасте.

Л.С. Выготский говорил, что в фантазиях подросток проживает свою богатую внутреннюю эмоциональную жизнь, свои порывы, что в фантазии он находит средства направления эмоциональной жизни. В творческих образах подросток воплощает свои эмоции, свои влечения.

В этом плане подросток осуществляет субъективную деятельность, обслуживает свою эмоциональную деятельность, однако его фантазии также могут выражаться в объективном творчестве. При создании чего-либо нового в процессе понимания или практической деятельности главным инструментом выступает фантазия. Именно в подростковом возрасте фантазия получает широкое развитие, в некотором смысле сближаясь с мышлением.

Обе стороны фантазии взаимосвязаны. Объективные выражения имеют оттенок эмоций, а субъективные фантазии часто наблюдаются в области объективного творчества. [5, 284-285 с] Как говорилось выше, при неправильно организованной учебной деятельности, можно получить достаточно скудные результаты, в полной мере не раскрыть потенциал учащихся, поэтому при

обучении программированию, следует делать также упор и на развитие творческих способной учащихся. Следует отметить, что для повышения мотивации учащихся следует не только выбрать правильную среду, но и выбрать правильные мотивационные инструменты и учитывать возрастные особенности учащихся [21].

Также следует сказать об одном из важных направлений в психическом развитии подростка, связанного с формированием способов преодоления проблем и трудностей. Выделяются 2 вида: конструктивные и неконструктивные способы. [24]

Конструктивные способы решения проблем направлены на активное преобразование ситуации, на преодоление травмирующих обстоятельств, в результате чего возникает чувство роста собственных возможностей, усиление себя как субъекта собственной жизни. Это вовсе не означает отсутствия тревог и сомнений в будущем.

Конструктивные способы:

- достижение цели собственными силами;
- обращение за помощью к другим людям, включенным в данную ситуацию или обладающим опытом разрешения подобных проблем;
- тщательное обдумывание проблемы и различных путей ее решения;
- изменение своего отношения к проблемной ситуации;
- изменения в себе самом, в системе собственных установок и привычных стереотипов).

Неконструктивные способы поведения направлены не на причину проблемы, которая «задвигается» на дальний план, а представляют собой различные формы самоуспокоения и выхода негативной энергии, создающие иллюзию относительного благополучия.

Неконструктивные способы:

– формы психологической защиты — вплоть до вытеснения проблемы из сознания;
– импульсивное поведение, эмоциональные срывы, экстравагантные поступки, необъяснимые объективными причинами;
– агрессивные реакции.

Исходя из вышесказанного следует вывод, что при обучение программированию в 5-6 классах следует учитывать следующие критерии при выборе средств обучения:

– Простота освоения среды программирования;
– Возможность работы в группах;
– Наличие возможностей развития творческих способностей для повышения мотивации при работе в среде.

1.2 Среды программирования используемые в курсе информатики и ИКТ

Программирование положительно влияет на развитие мышления. Для того, чтобы эффективно обучать учащихся 5-6 классов программированию, нам необходимо работать в понятной и интересной среде, легкой в усвоении с дружелюбным интерфейсом, позволяющим также проявлять творчество в деятельности.

Далеко не все средства программирования имеют это достоинство. Обычно в пропедевтическом курсе информатики для обучения основам алгоритмизации и программирования используются исполнители алгоритмов.

Исполнителями алгоритмов принято называть объекты или субъекты, для управления которыми составлен алгоритм. У любого исполнителя алгоритмов есть система команд исполнителя (СКИ), которая является совокупностью команд, которые исполнитель умеет выполнять. Также у каждого исполнителя есть определенная среда, или по-другому «место обитания» исполнителя.

Существует множество исполнителей, мы остановимся на трех исполнителях, которые используются чаще всего:

– Чертёжник. Предназначением данного исполнителя является построение рисунков, графиков, чертежей и т.д. Чертежник имеет перо, которое можно поднимать, опускать и перемещать. При перемещении опущенного пера за ним остается след – отрезок от старого положения пера до нового. Всего Чертежник умеет выполнять четыре команды: опустить перо, поднять перо, сместиться в точку (x, y), сместиться на вектор (a, b).
– Робот. Робот действует на прямоугольном клетчатом поле. Между некоторыми клетками могут быть расположены стены. Какие-то клетки могут быть закрашены. Сам Робот всегда занимает ровно одну клетку поля. Робот умеет выполнять всего 17 команд: 5 команд-приказов и 12 команд-вопросов. Команды – приказы: вверх, вниз, вправо, влево, закрасить. По командам вверх, вниз, влево, вправо Робот перемещается в соседнюю клетку в указанном направлении. Если на пути оказывается стена, команда не может быть выполнена. По команде закрасить Робот закрашивает клетку, в которой стоит. Если клетка уже была закрашена, она останется закрашенной, т.е. команда будет выполнена, но никаких видимых изменений не произойдет [7].
– Черепаха. Черепаха во многом похожа на чертежника. Разница в способе перемещения. В то время, как чертежнику необходимо задавать координаты, черепахе мы подаем команды вперед, назад и влево, вправо. Командам вперед и назад мы подаем число, отвечающее за количество шагов, которые следует пройти, а у команд влево и вправо мы определяем угол поворота [6].

Кроме определённой системы команд, для программирования исполнителей также используется алгоритмический язык программирования. Данный язык был введен академиком А.П. Ершовым в середине 1980-х годов.

Одним из главных дидактических достоинств исполнителей алгоритмов является наглядность среды исполнителей и их системы команд.

Говоря об обучении программированию в 5-6 классах нельзя не сказать о языке программирования Лого. Лого является языком программирования высокого уровня. Он был разработан в 1967 году Уолли Фёргезом, Сеймуром Пейпертом и Синтией Соломон для обучения детей дошкольного и младшего школьного возраста основным концепциям программирования [11].

Первое научное подтверждение успеха применения Лого как средства обучения было продемонстрировано Идит Харель, студенткой Пейперта, которая использовала Лого для обучения детей программированию и дробям в 1988 году [11]. На 6 марта 2009 года насчитывалась около 196 версий языка Лого [26], из чего можно сделать вывод, что данный язык оказал большое влияние на данный сектор сред программирования. Самой известной средой для изучения программирования в 5-6 классах на языке Лого является среда ЛогоМиры (MicroWorlds EX).

Одной из ключевых особенной данного семейства сред программирования является черепашья графика, которая служит наглядности при обучении программированию. Можно сказать, что этот принцип является основным для любой среды программирования для обучения школьников более младшей ступени, чем 7 класс, в чем мы убедимся на дальнейших примерах.

Среда программирования ЛогоМиры является многофункциональной инструментальной средой программирования. Данная среда содержит текстовый, графический, музыкальный редакторы, возможность записи звука с микрофона, набор программируемых объектов и сам язык программирования Лого [12].

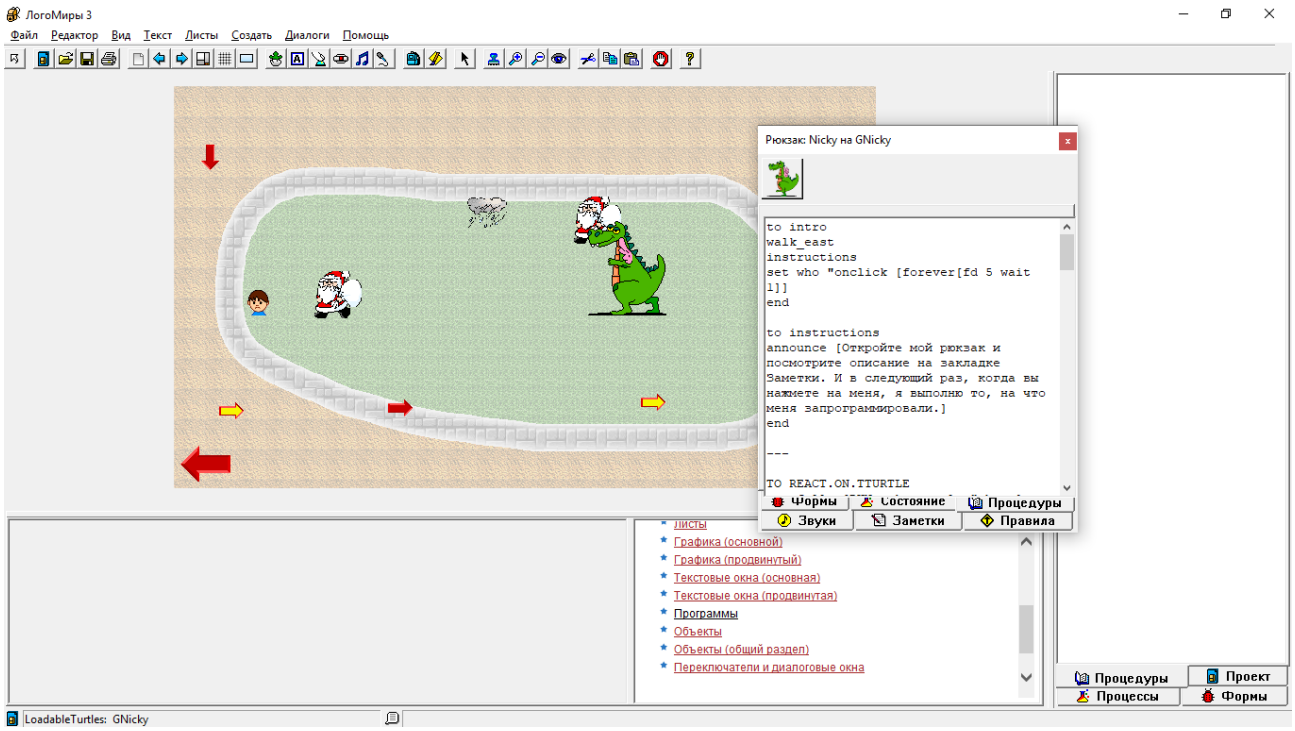

Рис. 1. Пример проекта в среде ЛогоМиры

Как можно увидеть на рисунке (Рис. 1), интерфейс среды ЛогоМиры не является самым простым из-за большого количества пиктограмм и меню, что может отразиться на освоения ее не только учениками 5-6 классов, но и преподавателями.

Действия каждого объекта программы задаются в «рюкзаке» объекта. В данном рюкзаке можно задать формы, которые может принимать объект; звуки, которые может использовать; процедуры, которые он может выполнять.

Сама по себе среда программирования является объектом, который можно запрограммировать. В примере (Рис. 3), можно увидеть скриншот программы, которая генерирует точки на белом фоне, которые могут отскакивать от черных стенок и уничтожаться при приближении к правым. На обоих примерах видно, что для программирования на данном языке требуется изучить его синтаксис.

Положительным моментом является то, что предоставляется внутренняя справка, которая помогает изучить как интерфейс программы, так и некоторые азы программирования на языке Лого. Также есть

возможность программировать на русском языке. Следует выделить то, что данная среда является платным продуктом, что может отвернуть от использования в некоторых школах.

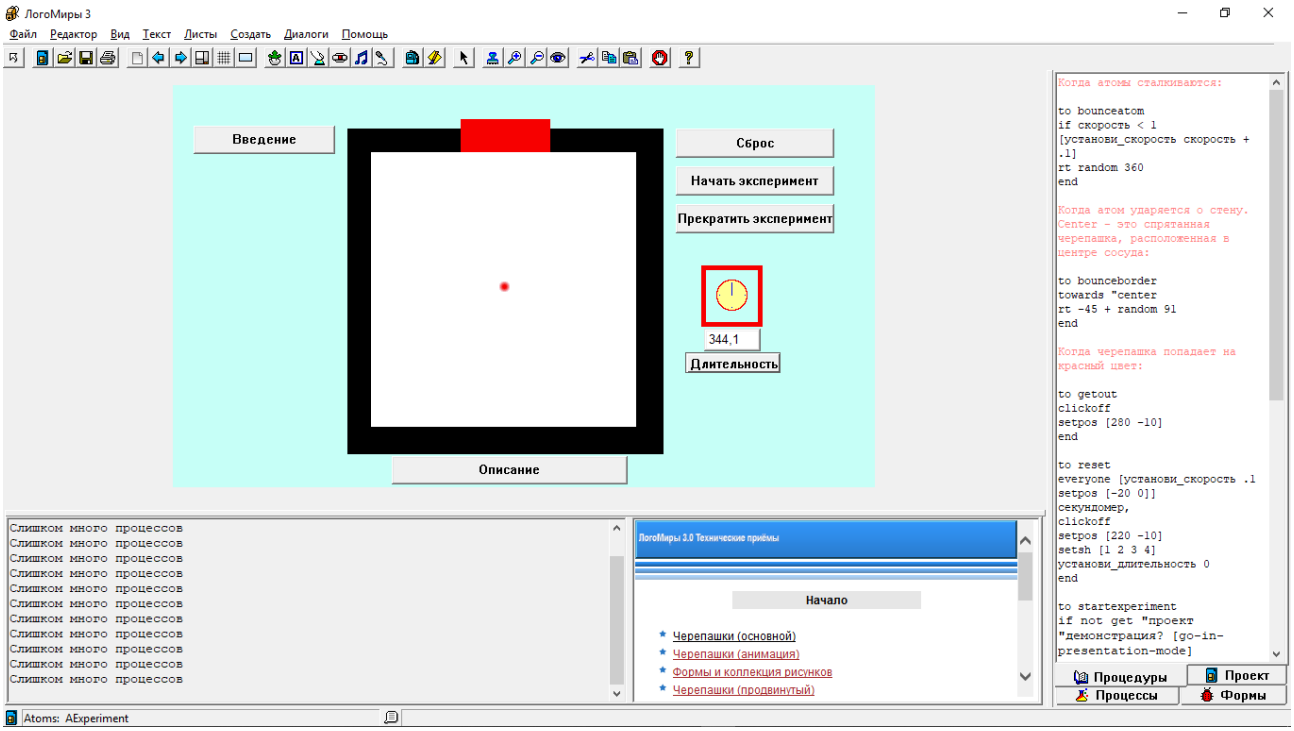

Рис.2. Пример программы в ЛогоМиры

Из всего вышесказанного выделим плюсы данной среды:

– относительная простота кода;
– программирование отдельных объектов и настройка их поведения;
– встроенный графический, музыкальный и текстовый редакторы.

Из минусов можно выделить:

– устаревший и сложный интерфейс;
– распространяется платно.

Среда программирования КуМир, или Комплект учебных Миров была разработана в ФГУ ФНЦ НИИСИ РАН по заказу Российской академии наук [9]. Данная среда программирования была разработана, основываясь на методике А.П. Ершова, предложенной в 1980-х годах. Эта методика широко использовалась в средних школах СССР и России [8].

Данная среда программирования свободно распространяется. В текущей версии КуМир используются исполнители «Водолей», «Робот», «Рисователь», «Вертун», «Чертежник».

Для исполнителей «Водолей», «Робот» и «Вертун» есть пульты исполнителя, которые позволяют выполнять алгоритмы, не прибегая к программированию. Также есть встроенные руководство пользователя и практические задания.

Говоря о программировании, следует сказать, что оно осуществляется на русском языке, компилятор проверяет текст программы на ошибки. Есть возможность вставки некоторых алгоритмических конструкций, как циклы или условия, ускоряя процесс написания кода. К сожалению, пользователь данной среды программирует поведение определённых исполнителей, что исключает создание творческих проектов.

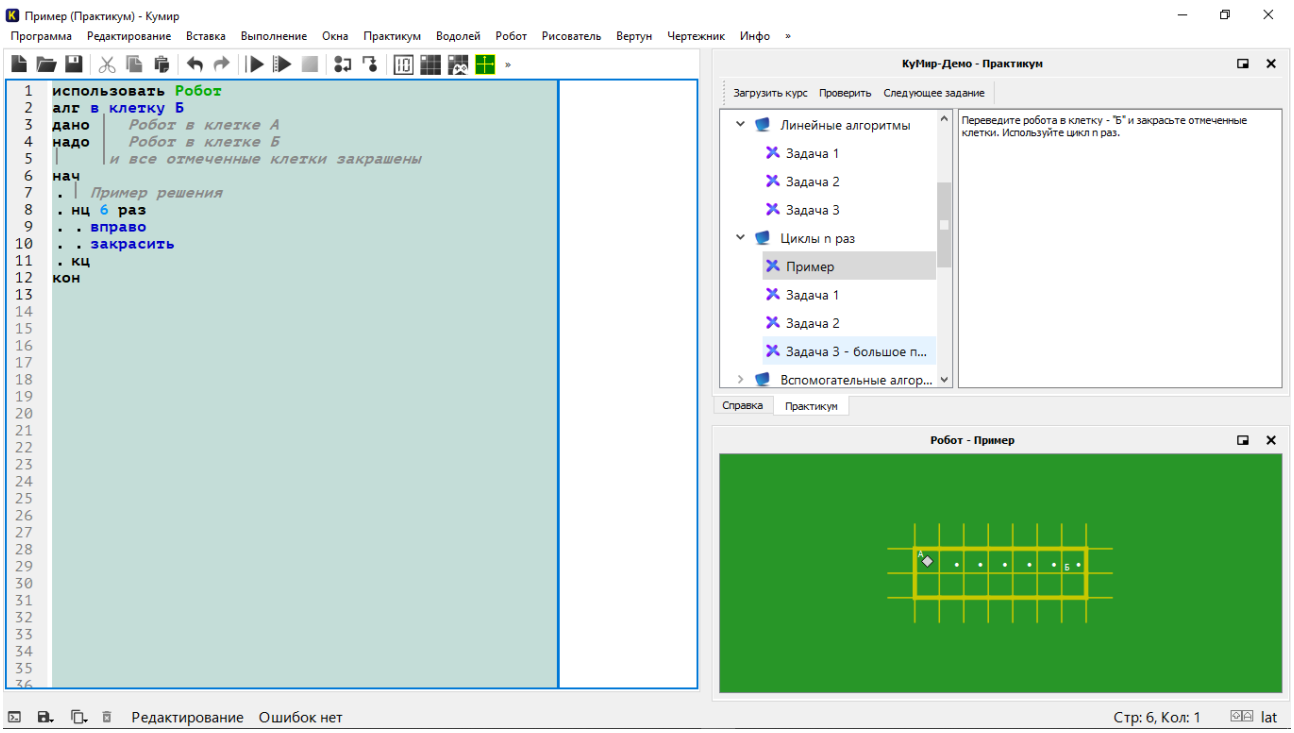

Рис.3. Пример программы в КуМир

Подводя итог, можно выделить следующие плюсы:

– относительная простота кода;

– легкий интерфейс;

– встроенное руководство и набор практикумов;

– наличие средств исполнения алгоритмов, без необходимости программирования;
– бесплатность.

Из минусов можно выделить лишь возможность программировать исключительно исполнители алгоритмов, не имея возможности создавать собственные проекты. Это не позволят в полной мере раскрыть потенциал, в том числе и творческий, а также не дадут возможность познакомится с другими направлениями в программировании. [16]

1.3 Среда программирования Scratch

Среда программирования Scratch была разработана группой Lifelong Kindergarten из Массачусетского технологического университета, которую возглавлял Митчелл Резник. Первая версия Scratch была выпущена ими в 2003 году, и целью этой программы было помочь молодым людям от 8 и старше изучать программирование.

Scratch имеет онлайн-сообщество, в котором, на 2017 год, было зарегистрировано более 22 миллионов пользователей. Также среда поддерживает более 70 языков, в том числе и русский. В январе 2019 года была выпущена Scratch 3.0, о которой мы и будем говорить в данной работе [19].

Scratch является визуальной событийно-ориентированной средой программирования для школьников младших и средних классов. Программирование в данной среде происходит при помощи блоков, которые сцепляются друг с другом как конструктор Lego. Основной программируемой единицей выступают спрайты. Перемещать элементы интерфейса, в которые входят блоки и спрайты, можно при помощи метода Drag and Drop.

Среда программирования Scratch обладает графическим, текстовым и звуковым редактором. Можно проследить, что сама среда много почерпнула

от Лого и ЛогоМиров, учитывая, что создателем Лого и Scratch был Сеймур Пейперт.

Как и у редакторов, рассмотренных выше, данная среда обладает встроенным руководством пользователя, где пользователей учат использованию языка при помощи ряда видеороликов, где ученики воспроизводят проекты из обучающего видео.

Несомненным преимуществом, котором обладает данная среда является её относительная мобильность. Есть возможность запускать Scratch как используя установленный дистрибутив, так и запустив редактор в браузере. Кроме того, следует сказать о том, что данная среда свободно распространяется.

Из плюсов данной среды программирования можно выделить:

- Простой в освоении язык
- Современный и красочный интерфейс
- Поддерживаемое официальное онлайн-сообщество
- Наличие встроенного графического, текстового и музыкального редакторов
- Свободно распространяемое ПО
- Наличие встроенного руководства

Рис.4. Пример программы в Scratch

Единственным минусом можно назвать неудобно построенную работу с переменными и условиями, которые могут получиться слишком громоздкими. [16]

Scratch является визуальной событийно-ориентированной средой программирования для школьников младших и средних классов. Целью создания этого языка было позволить детям, у которых нет опыта программирования, изучить основные принципы императивного, объектно-ориентированного и многопоточного программирования. Этот язык удобно использовать как начальный язык программирования, потому что создавать проекты достаточно легко, а полученные навыки могут применяться в таких языках, как Python и Java [8], т.к. в Scratch присутствуют:

- стандартные для языков процедурного типа: следование, ветвление, циклы, переменные, типы данных (целые и вещественные числа, строки, логические, списки – динамические массивы), псевдослучайные числа;
- объектно-ориентированные средства: объекты (их поля и методы), передача сообщений и обработка событий;
- интерактивные средства: обработка взаимодействия объектов между собой, с пользователем, а также событий вне компьютера (при помощи подключаемого сенсорного блока);
- средства параллельного выполнения: запуск методов объектов в параллельных потоках с возможностью координации и синхронизации;
- создание простого интерфейса пользователя [13]

Кроме того, Scratch обладает следующими уникальными особенностями:

- **Блочное программирование**. Для создания программ в Scratch пользователь совмещает графические блоки, которые мы описали в параграфе выше. Особенностью данных блоков является то, что различные блоки имеют разные формы, соответственно блоки

можно собрать только в синтаксически верных конструкциях, что исключает ошибки. Также в созданном скрипте возможно делать изменения во время работы программы, что позволяет пользователям проводить дополнительные эксперименты со своими программами.

- **Манипуляции данными.** В Scratch пользователь может работать с текстовой, графической и звуковой информацией, создавать музыку и анимацию, смешивать их. Например, пользователь может добавить программируемость, похожих на Photoshop, фильтров. [13].

Выше было сказано, что Scratch кроме самой среды является и онлайн-площадкой, где можно делиться проектами и обсуждать их. В личном профиле Scratch есть возможность создания собственной студии.

В студии есть возможность создавать новые проекты или добавлять уже существующие проекты в студию. Делать это могут менеджеры. Изначально менеджером является создатель студии. Создатель может объявить куратором студии пользователей, которые следят за студией или проектами студии. Если куратор будет повышен до менеджера, у него будет возможность добавлять свои проекты и редактировать существующие.

Редактирование одного проекта двумя пользователями не синхронное. Однако они могут распределить над какими частями проекта они работают. После сохранения проектов обоими участниками и новой загрузки этого проекта, все части проекта будут синхронизированы.

Также скажем несколько слов о проектах, которые возможно разрабатывать в Scratch. Разберем основные типы проектов, которые реализуются в Scratch:

- Анимации, один из популярных типов проектов. Характеризуются содержанием определенного числа «костюмов», играющих определенную анимацию в быстрой последовательности для того,

чтобы создать фильм или что-то похожее. Создаются для того, чтобы показать жизнь изо дня в день, фантастические истории, комедии и т.д.

- Игры являются самым распространённым типом проектов. Благодаря возможностям Scratch, были воссозданы такие игры, как Pacman или Mario.
- Модели. Несмотря на то, что симуляции редки, но большинство представленных проектов крайне высокого качества. Существуют работы, представляющие физику, погоду, гравитацию, 3D модели и т.д.
- Музыкальные проекты. Scratch использует звуковой банк MIDI, который позволяет пользователям воспроизводить до 128 инструментов, что позволяет играть ноты с определенной темой при этом регулируя громкость и темп.
- Художественные проекты, есть проекты, основная цель которых показать иллюстрации, в независимости от того, интерактивные они или нет. Несмотря на то, что в таких проектах обычно мало программирования или его нет вовсе, такие проекты также приветствуются командой Scratch, т.к. Scratch не только о программировании, но и о выражении творчества любыми способами.
- Истории. Таких проектов представлено не много, т.к. большинство приписываются в категорию анимационных проектов. Ранжируются от интерактивных историй, где пользователи создают собственную историю до шоу, где персонажи просто разговаривают друг с другом [27].

Рассмотрев особенности среды программирования и основные типы проектов, которые реализуются в Scratch, мы можем сделать вывод, что все проекты, так или иначе, включают в себя создание скриптов из различных

блоков, а также использование графики. Поэтому нам необходимо сравнить этот вышеназванные особенности с другими средами. Эти данные можно будет увидеть в Таблице 1.

Таблица 1.

Сравнение сред программирования

Параметр	ЛогоМиры	КуМир	Scratch
Простота освоения языка	-	+	+
Простота интерфейса	-	+	+
Наличие официального онлайн-сообщества	-	-	+
Наличие встроенных редакторов	+	-	+
Наличие руководства пользователя	+	+	+
Наличие встроенных практикумов	-	+	-
Свободно распространяемое ПО	-	+	+

Следует уделить несколько слов существующей методической литературе по Scratch. В первую очередь следует сказать о «Книге юных программистов» за авторством Голикова Д. В этой книге автор составил ряд лабораторных работ, которые учат программировать в среде Scratch. Данное руководство ориентировано на учеников 3-5 классов. [1]

Также стоит сказать про учебное пособие «Среда программирования Scratch» от Боровича П.С. и Бутко Е.Ю. в этом пособии также приведен ряд лабораторных работ [1]. Кроме того, можно найти много видеороликов с похожим содержанием на площадке YouTube.

Разработчики позиционируют Scratch, как среду для создания различных игр, историй, где оживают созданные детьми персонажи, где первоочередной задачей ставится реализация детьми какой-либо истории, а уже затем обучение программированию. Поэтому хорошим вариантом заданий для учащихся будет постановка развлекательно-образовательной задачи на уроках, в которой учащиеся могли бы реализовывать собственные задумки и учиться программировать.

Глава 2. Исследовательская часть

2.1 Разработка учебных проектов в визуальной событийно-ориентированной среде программирования Scratch

В рамках исследовательской работы были реализованы пять проектов в среде программирования Scratch по следующим темам:

1. Питомец – аналог игры Тамагочи или «Кот Том». Игроку требуется кормить, поить и укладывать спать питомца. Для реализации проекта использовались основные алгоритмические конструкции – циклы, условные операторы, переменные и некоторые особенности самих блоков Scratch.
2. Арканоид – клон одноименной игры. Игроку необходимо при помощи мышки управлять платформой и отбивать мяч, который должен сбить кирпичи, летящие сверху. Для реализации проекта использовались основные алгоритмические структуры и работа с сенсорами, например, управление платформой при помощи компьютерной мыши.
3. Полет – клон игры FlappyBird. Игроку необходимо лететь сквозь город и стараться не прикасаться к зданиям.
4. Прыжки – в данной игре необходимо перепрыгивать через препятствия.
5. Догонялки – в данной игре необходимо определенное количество раз догнать предмет, который каждый раз выскальзывает из рук игрока и оказывается в случайном месте на поле. В некоторой степени проект можно считать, как проект повышенной сложности, т.к. кроме стандартных алгоритмических конструкций используются также методы.

Разберем подробнее возможное выполнение каждого проекта. Методические рекомендации, которые будут включать тексты заданий будут приведены во второй части главы.

Проект «Питомец». Разберем один из способов выполнения этого проекта. Читая задание, мы можем выделить несколько задач, которые нам необходимо реализовать в проекте, а именно реализовать процессы сна, питания, питья и придумать способ развлечения нашего питомца. Это значит, что нам как минимум требуется пять спрайтов, одним будет управлять игрок, а с оставшимися он будет взаимодействовать.

В данном варианте проекта нашим питомцем является кот, а в качестве развлечения ему служит прослушивание музыки, живет же он в просторной комнате с кроватью.

Рис. 1. Сцена проекта

Определившись с необходимыми для нашего проекта спрайтами, мы можем приступить к кодированию. Для начала нам потребуется три переменные для отслеживания состояний сна, сытости и жажды нашего питомца, которые постоянно уменьшаются в процессе работы программы.

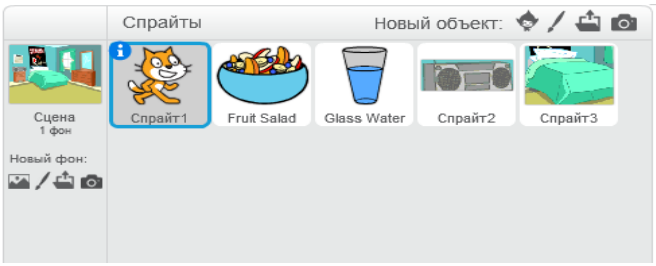

Рис. 2. Необходимые в проекте спрайты

Основная идея программы заключается в том, что мы, при нажатии на каждый спрайт посылаем сообщение о сделанном действии, т.е. когда мы кликнем на спрайт с водой, нам отправится сообщение «Пить». Когда игрок получает сообщение «Пить», его спрайт подходит к стакану с водой и пьет из него, т.е. восстанавливает определённое количество переменной «Вода». По аналогии мы поступаем и с остальными переменными, за исключением работы радио. Когда мы кликаем на радио мы не посылаем сообщение, но просто начинаем играть музыку.

Также нельзя доводить нашего питомца до ожирения, поэтому если значение трех основных переменных является выше определённого программистом уровня, то наш питомец должен как-то подавать знак того, что он не голоден и наоборот, если у какого-либо из трех основных переменных низкие значения наш питомец должен подать знак того, чтобы его накормили, напоили или уложили спать.

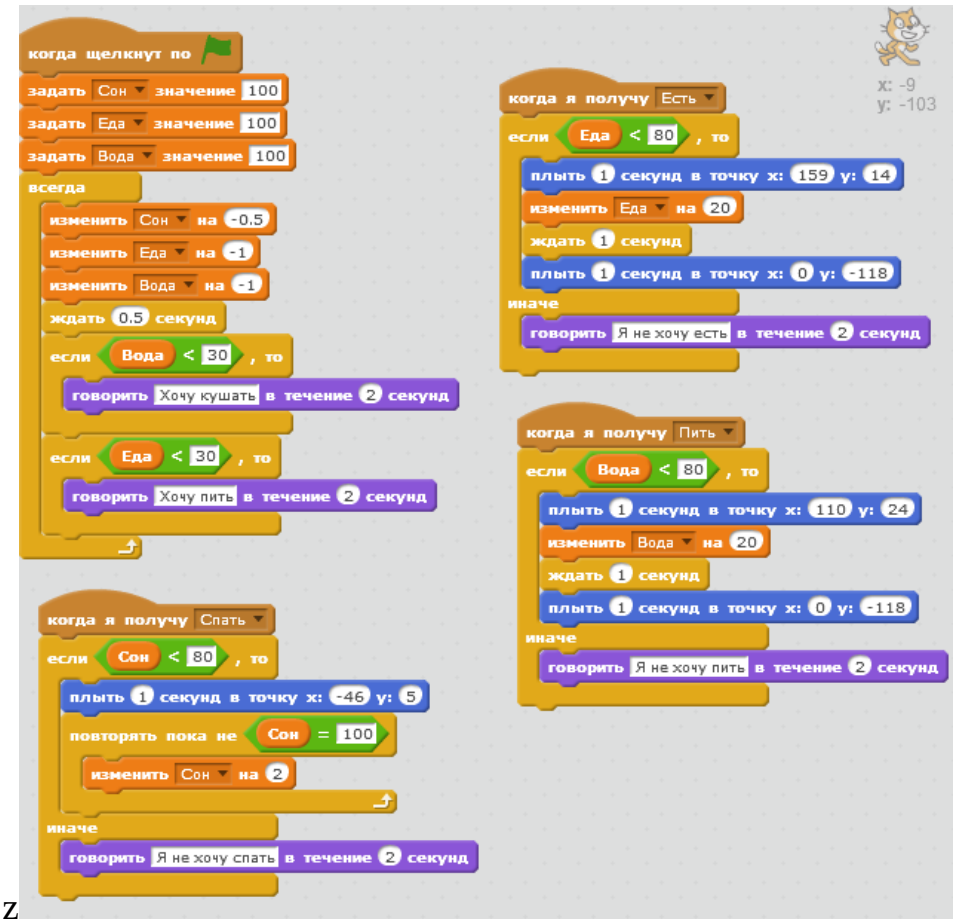

Рис. 3. Код спрайта "Спрайт1"

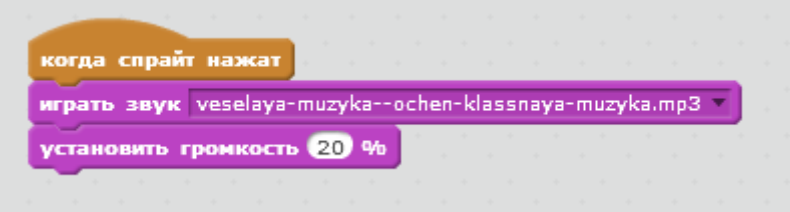

Рис. 7. Код спрайта "Спрайт 2"

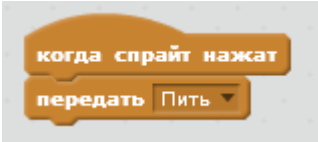

Рис. 6. Код спрайта "Glass Water"

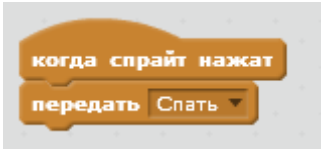

Рис. 5. Код спрайта "Спрайт 3"

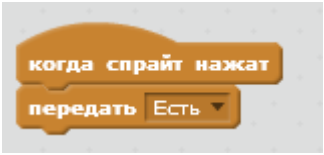

Рис. 4. Код спрайта "Fruit Salad"

Данный вариант выполнения проекта лишь один из многих. В зависимости от уровня учеников в классе данный проект можно предложить для выполнения, как и индивидуально, так и как в качестве группового проекта.

Проект «Прыжки». В задании сказано, что нам необходимо преодолевать препятствия, возникающие на пути, т.е. предполагается движение игрока, однако в данном случае мы реализуем не движение, а лишь иллюзию движения. Для этого мы должны анимировать спрайт, которым будет управлять игрок. В библиотеке Scratch уже есть объекты с анимацией, так что нам не потребуется выполнять дополнительную работу.

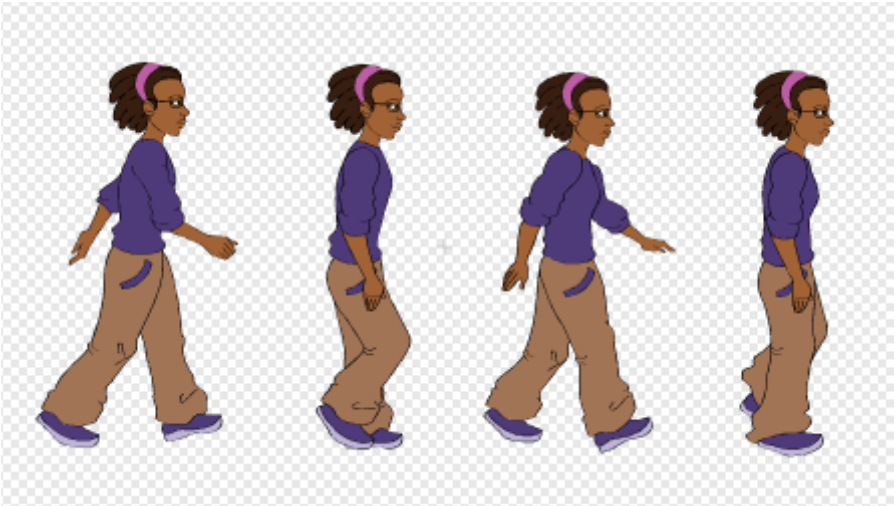

Рис. 8. Костюмы движения персонажа Avery

Однако придание спрайты анимации движения недостаточно. Если мы не можем идти к препятствиям и перепрыгивать их, то препятствия должны

прийти к нам. Учащиеся могут выбрать из библиотеки любое понравившийся им объект в качестве препятствия, мы же выберем в качестве препятствия дерево. Также для того, чтобы иллюзия движения выглядела правдоподобнее, добавим объект облако.

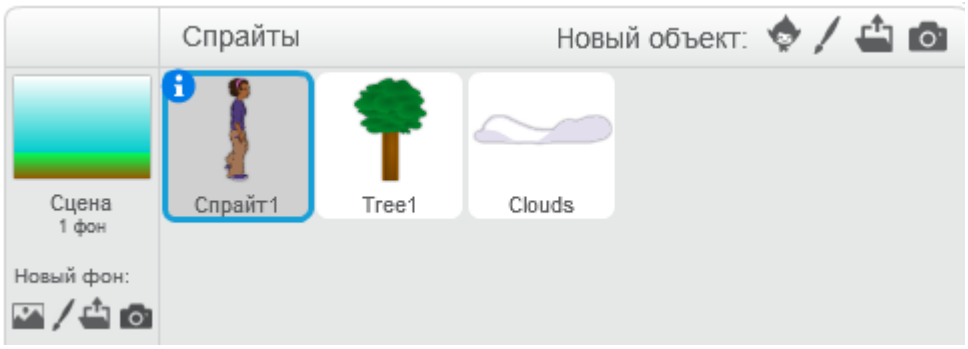

Рис. 9. Спрайты проекта

Однако добавление одного дерева и облако явно недостаточно, т.к. игра должна быть достаточно продолжительна, чтобы она не заканчивалась на том, что игрок перепрыгнул одно единственное дерево. Поэтому мы будем использовать клонирование объектов, а затем уничтожение этих клонов, после того, как мы их перепрыгнем.

Идея заключается в том, что оба объекта – дерево и облако – спрятаны и не видны игроку в самом начале, однако после того, как игра начнется дерево и облако скопируют себя и после этого игрок будет видеть клоны этих объектов, которые будут уничтожены, дойдя до края карты. Во время создания клона требуется подождать некоторое время, чтобы препятствия не генерировались постоянно и у нас была возможность их обойти.

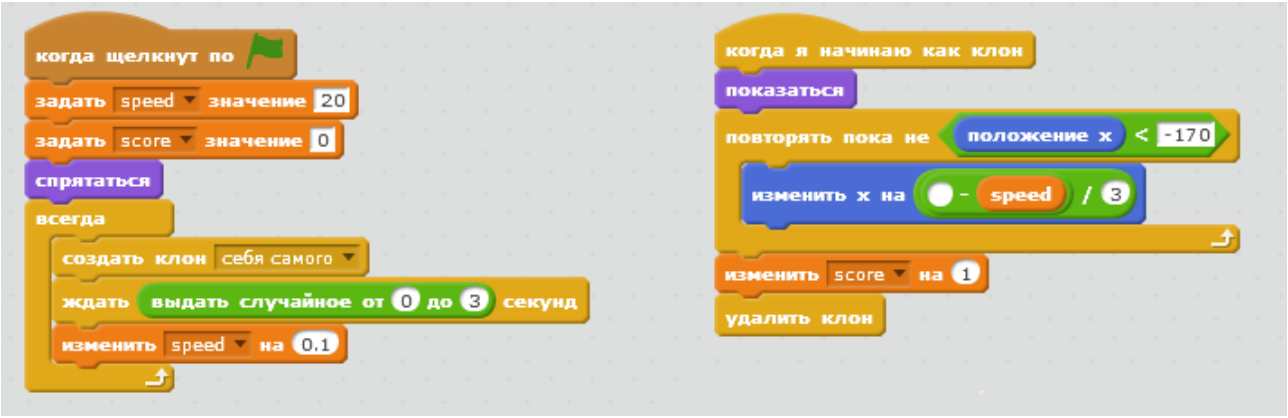

Рис. 10. Скрипт спрайта "Tree1"

Разница между ними состоит в том, что препятствия набирают скорость и чем дольше идет игра, тем быстрее они надвигаются на игрока. Чтобы препятствия были разнообразнее можно добавить им несколько костюмов, которые будут меняться случайным образом, или создавать одновременно несколько препятствий, т.е. убирать задержку между клонированием.

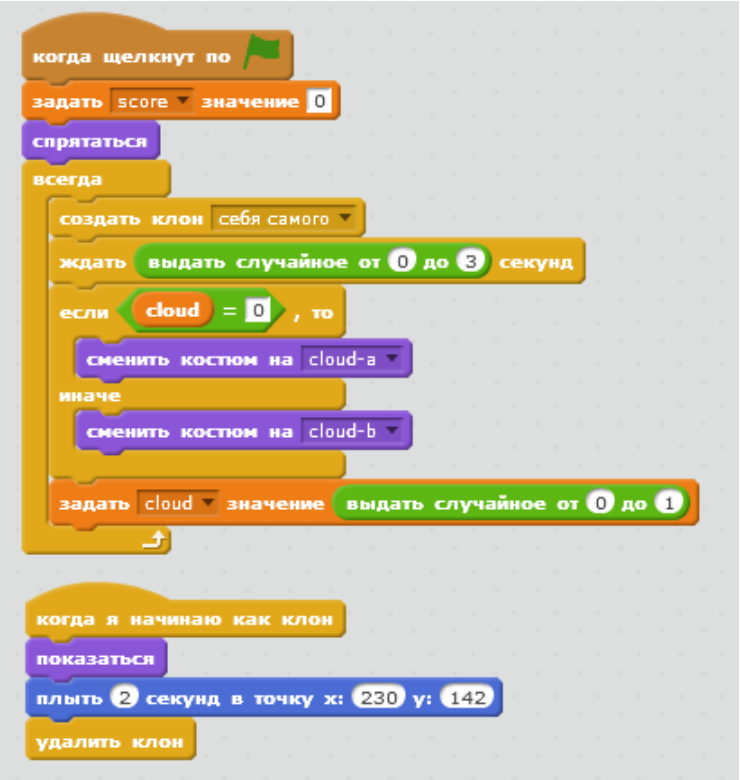

Рис. 11. Скрипт спрайта "Clouds"

Также важно сказать про движение игрока. Как уже было сказано выше, мы создаем лишь иллюзию движения. У нас есть анимация движения, теперь нам необходимо реализовать действия для персонажа, которым мы будем управлять. Учитывая, что мы создаем иллюзию движения наш персонаж будет лишь прыгать, чтобы преодолевать препятствия.

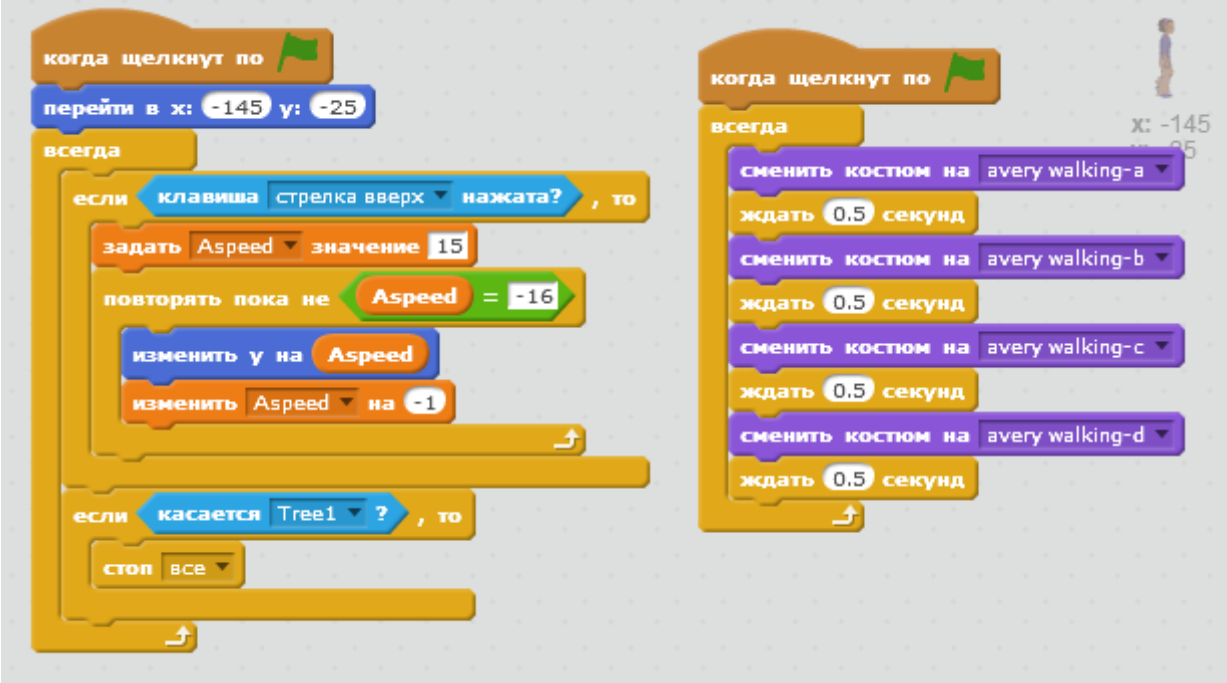

Рис. 12. Скрипты спрайта "Спрайт 1"

Проект «Полет». В нашем варианте работы мы будем управлять летучей мышью и перелетать здания. Генерация зданий происходит точно также, как и в предыдущем проекте. Разница заключается в том, что каждому клону присевается случайная координата Y при генерации. Это позволит каждому зданию быть разной высоты.

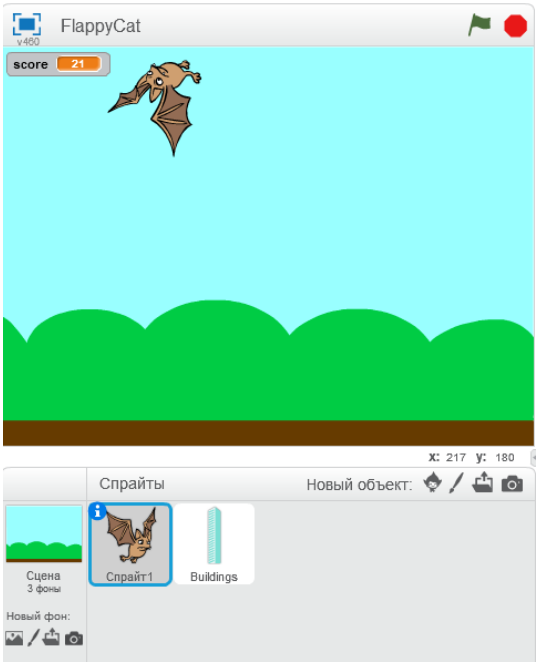

Рис. 13. Сцена и спрайты проекта

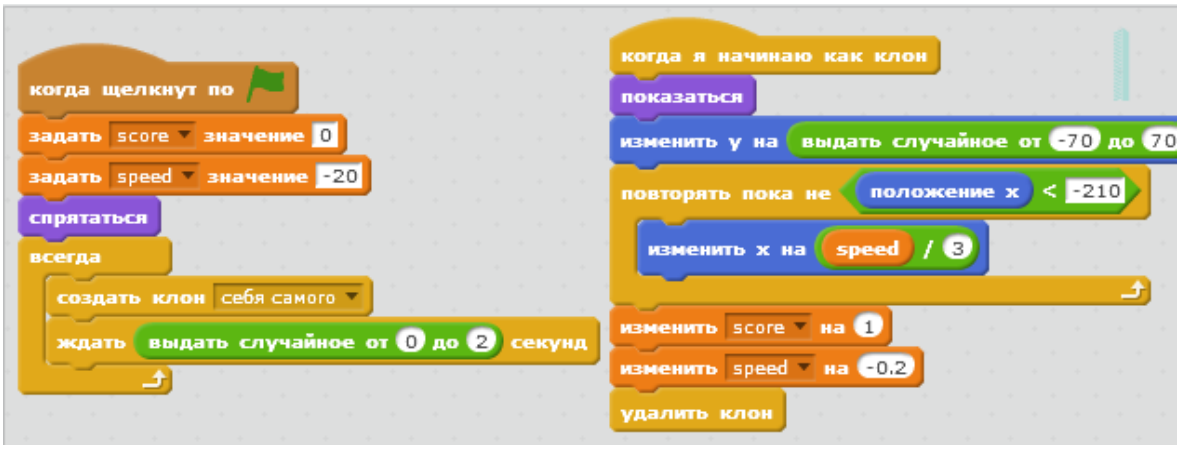

Рис. 14. Код спрайта "Buildings"

Данная игра похожа на предыдущий проект. Отличие от предыдущего проекта заключается в том, что наш персонаж постоянно падает, это значит, что нам нужно постоянно взлетать и уклоняться от препятствий. В остальном же мы точно также стараемся не попадать на препятствия и набираем очки за каждое здание, которое перелетели.

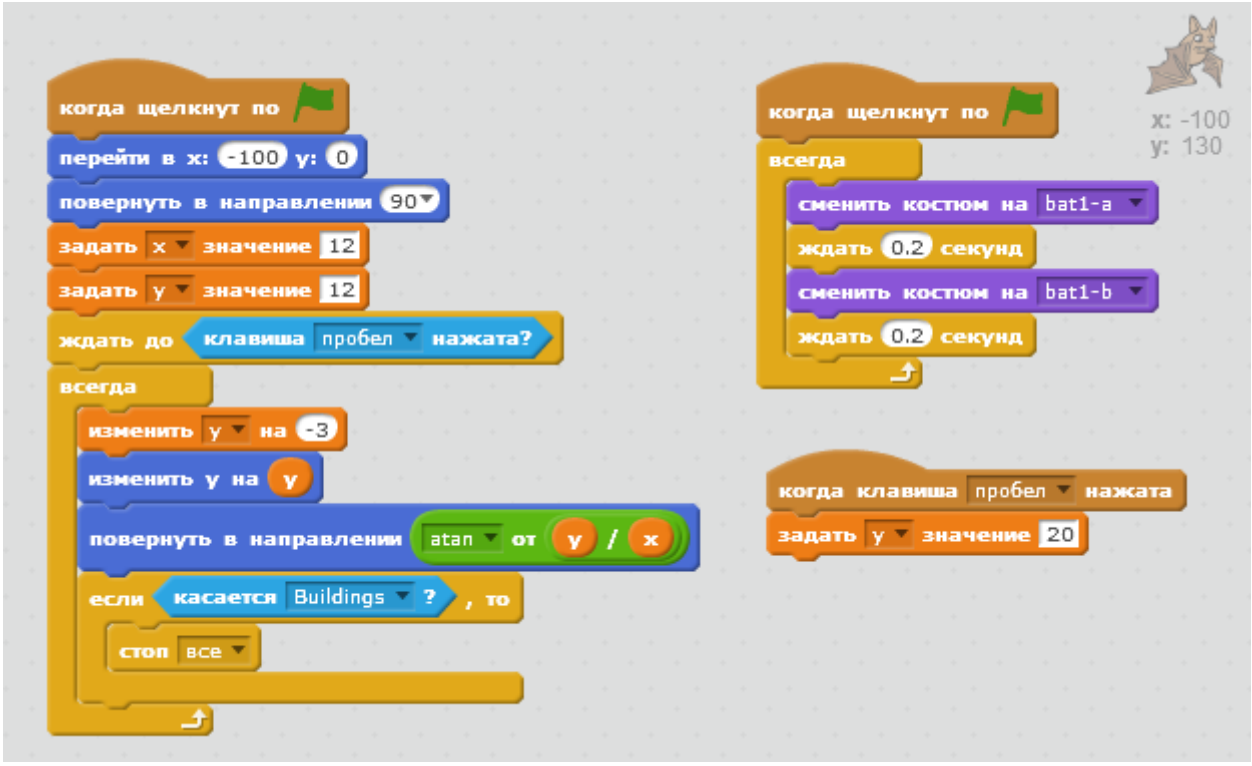

Рис. 15. Скрипт летучей мыши

Проект «Арканоид». В этой игре нам будет необходимо управлять платформой. Чтобы это сделать, мы скажем, что будем перемещать платформу туда, куда будет указывать мышь. Так как мы должны двигаться исключительно по оси X, зафиксируем координату Y в определенное значение.

Шарик всегда генерируется из центра сцены. Чтобы он летел вниз, нам необходимо, чтобы у него уже была со старта задана определенная скорость, и чтобы он был направлен в определенную сторону.

Как только мы касаемся платформы, шарик должен повернуться на 180 градусов, т.е. в противоположную сторону. Чтобы игра стала интереснее, необходимо повышать скорость шарика на определенное значение, тогда он с каждым разом будет двигаться быстрее и отбить его будет труднее. Также, чтобы предотвратить пропажу шарика за пределы сцены, добавим условие «если на краю, оттолкнуться».

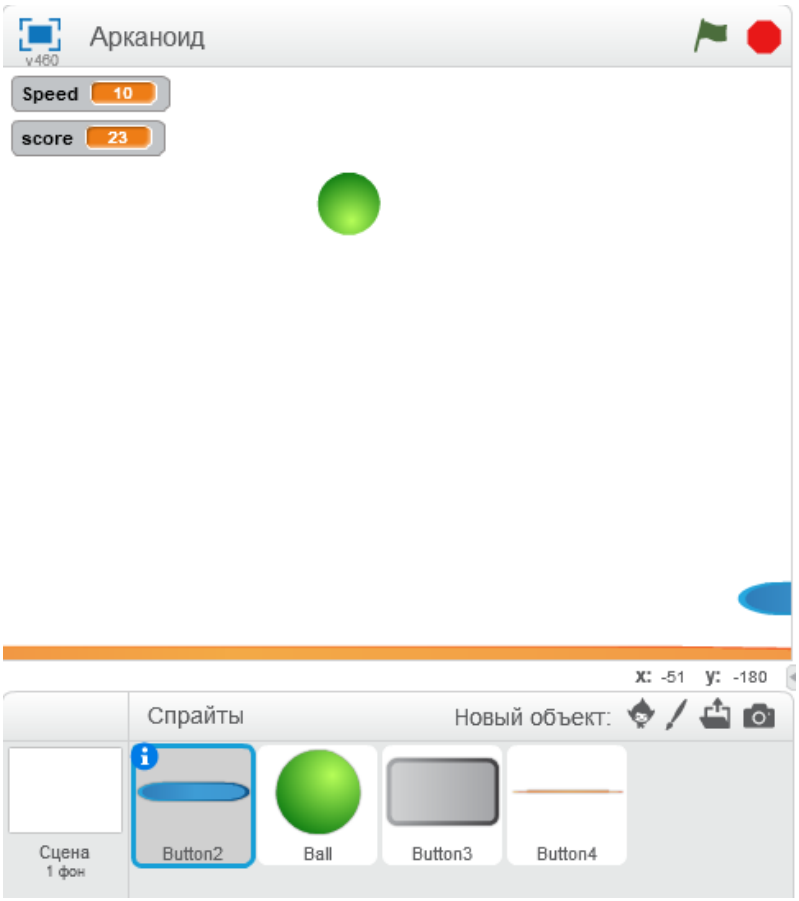

Рис. 16. Сцена и спрайты игры

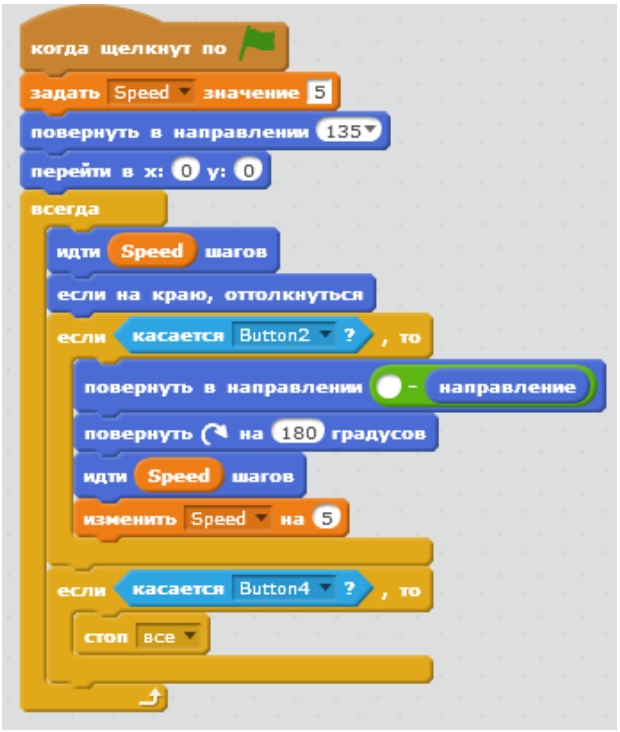

Рис. 17. Скрипт спрайта "Ball"

Осталось лишь начать генерацию кирпичей. Мы также воспользуемся клонированием, но ждать в этот раз не будем. Вместо этого, мы будем повторять генерацию клонов n раз и задавать им случайные координаты вверху сцены. При попадании должен срабатывать сенсор касания о шарик. Если было попадание, то начисляются очки.

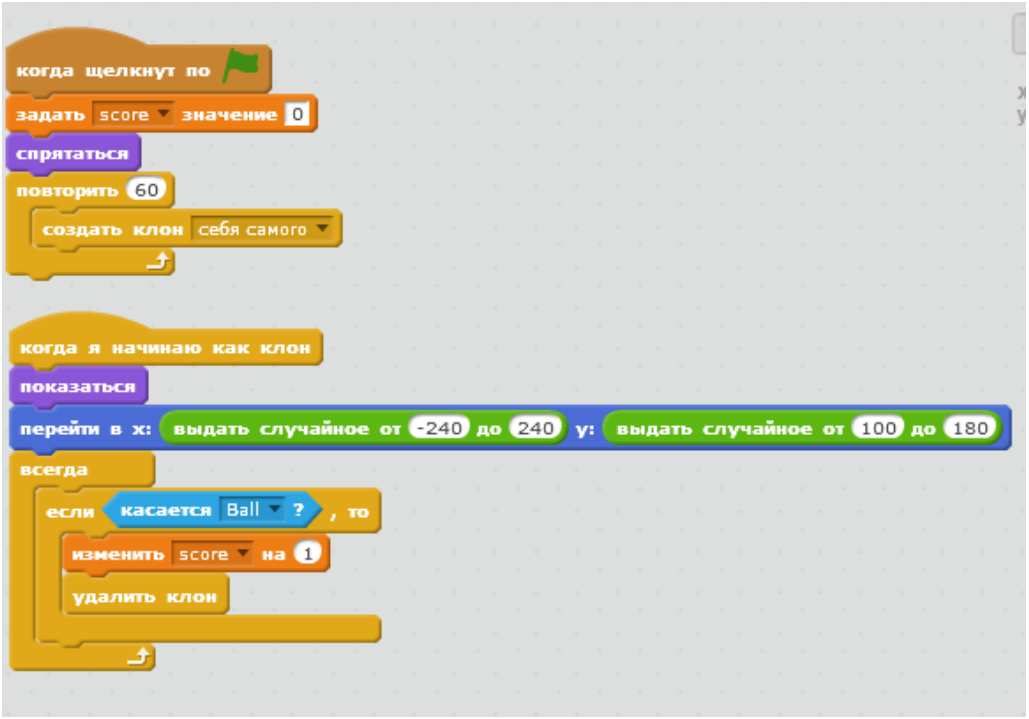

Рис. 18. Скрипт спрайта "Button3"

Проект «Догонялки». Для начала нам необходимо будет подготовить все необходимые для нашей игры спрайты. В данном варианте игры используется стандартный спрайт кота и спрайт банана с костюмом банана соответственно и блюда с фруктами, который появляется в конце вместо банана. Также необходимо будет подготовить сцену (фон), который можно будет как выбрать из библиотеки, так и нарисовать самостоятельно. Подготовив спрайты, начнем писать скрипты для наших спрайтов.

Рис. 19. Сцена и спрайты игры

Раз смысл игры догонять и в нашем случае банан, то игроку предстоит перемещаться по сцене, чтобы сделать это. Для того, чтобы перемещаться вправо, например, нам необходимо взять блок "Если" в группе "Управление" и задать условие из группы "Сенсоры" "Клавиша нажата?". Проверять мы будем нажатие клавиши "Стрелка вправо". Если условие выполняется, то тогда мы

перемещаем спрайт в положительном направлении оси абсцисс. Однако нам необходимо будет реализовать перемещение во всех четырех направлениях. Это можно сделать несколькими способами.

Первый способ – продублировать скрипт движения вправо и изменить значения на соответствующие, т.е. при движении влево мы перемещаем спрайт в отрицательном направлении оси абсцисс. Второй способ заключается в том, чтобы создать функцию с формальным логическим параметром. Мы также дублируем созданный скрипт, меняем соответствующие значения. Однако в основном скрипте спрайта мы вызываем только функцию.

Попробуем наступить на банан. Можно увидеть, что игрок проходит сквозь него, но ничего больше. Поэтому нам необходимо еще одно условие. По которому мы будем определять – наступил на банан игрок или нет. Нам также необходимо взять блок "Если" в группе "Управление" и задать условие из группы "Сенсоры" "Касается …?" и в списке выбрать необходимый нам спрайт. Также нам необходимо объявить переменную, для того, чтобы отслеживать сколько раз игрок поймал спрайт. Если условие выполняется, то мы изменяем значение переменной на единицу.

Теперь нам необходимо написать скрипт и для спрайта "Bananas", чтобы он мог "убегать" от игрока. Возьмем блок "Если" в группе "Управление" и задать условие из группы "Сенсоры" "Касается …?" и в списке выберем необходимый нам спрайт. Если условие выполняется, то зададим этому спрайту случайные координаты и если наш спрайт достиг края сцены, то пускай он оттолкнется от края.

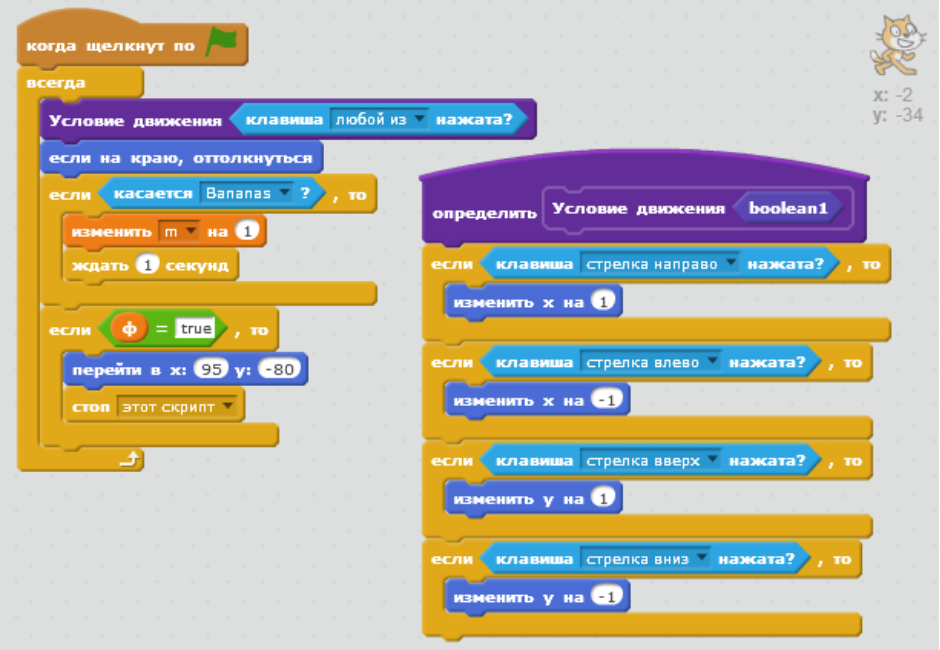

Рис. 20. Скрипты Спрайта1

Теперь почти все сделано, осталось добавить условие окончание работы скрипта. Выберем в группе операторы "<Условие 1> и <Условие 2>". У нас будет два условия окончания скрипта, первое условие – это достижение переменной m числа 10, т.е. игрок 10 раз коснулся спрайт "Bananas", а второе условие – это если у банана активирован первый костюм. При условии выполнения требований, мы меняем костюм банана на костюм тарелки с фруктами, устанавливаем её в центр сцены, т.е. в координаты (0;0). Также нам необходимо объявить переменную, в данном случае названную ф, которая будет отслеживать то, окончена игра или нет. Далее обнуляем значения касаний и устанавливаем переменную, которая отвечала за отслеживание того, окончена игра или нет в значение true (истина) и в блоке "Управление" выбираем блок "Стоп" и выбираем в списке "Этот скрипт". Переходим обратно к спрайту игрока, задаем условие "Если ф = true", то мы также останавливаем этот скрипт.

Осталось только получившиеся скрипты добавить в цикл "Всегда", который находится в блоке "Управление", чтобы команды скриптов не выполнялись единожды. Также в скрипте "Bananas" перед циклом зададим переменной ф значение false и создадим условие "Если выбран костюм 2 (тарелка с фруктами)", то тогда мы меняем костюм на банан. Это мы делаем

для того, чтобы не догонять в следующие раз тарелку с фруктами вместо банана.

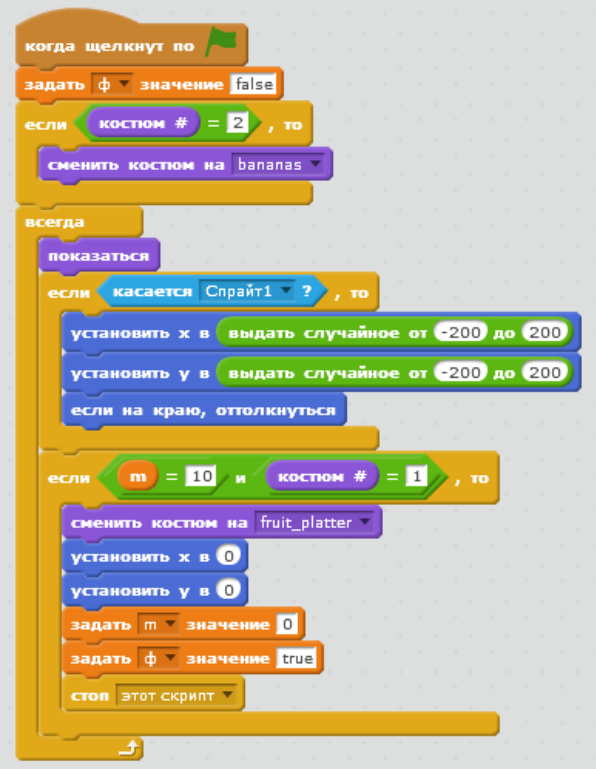

Рис. 21. Скрипт спрайта Bananas

2.2 Методические рекомендации для учителей

Проекты возможны к выполнению как индивидуально, так и в группах по 2-3 человека, что является более предпочтительным вариантом. Если ученики выполняют проект в группе, то предполагается, что каждый пишет скрипт для собственного спрайта. Среда Scratch ориентирована на творчество, а значит ученики могут быть организованы в группы, где один программирует, а другой ученик ищет необходимые ресурсы в сети или же рисует сам.

Задача проекта «Питомец»: разработать игру, в ходе которой игрок должен ухаживать за питомцем, а именно давать ему еду, воду, укладывать спать. Если питомцу очень долго не давали еду или воду, то высвечивается надпись, говорящая о безответственности игрока. Если питомец долго не спал, то он засыпает прямо на месте и не может ничего делать в течении 10 секунд. Также необходимо реализовать развлечение для кота, например, предоставить ему возможность слушать музыку.

Выполнение данного проекта учениками может преследовать несколько целей:

- Обучающая: ближе познакомятся с понятием переменных, условных операторов, познакомятся с некоторыми объектно-ориентированными средствами.
- Воспитательная: выполнение данного проекта может служить воспитанию гуманности в учениках.
- Развивающая: выполнение данного проекта может способствовать развитию логического и творческого мышления у учащихся.

Задача проекта «Прыжки»: разработать игру, в которой игрок должен перепрыгивать возникающие у него на пути препятствия. За каждое преодоленное препятствие будут начисляться очки. Если препятствие было задето, то игра начинается заново.

Выполнение данного проекта учениками может преследовать несколько целей:

- Обучающая: ближе познакомятся с понятием переменных, условных операторов, познакомятся с некоторыми объектно-ориентированными средствами, клонированием объектов;
- Развивающая: выполнение данного проекта может способствовать развитию логического и творческого мышления у учащихся.

Задача проекта «Полет»: разработать игру, в ходе которой игроку предстоит пролетать над возникающими на пути препятствиями. За каждое преодоленное препятствие будут начисляться очки. Если препятствие было задето, то игра начинается заново.

Выполнение данного проекта учениками может преследовать несколько целей:

- Обучающая: ближе познакомятся с понятием переменных, условных операторов, познакомятся с некоторыми объектно-ориентированными средствами, клонированием объектов;

– Развивающая: выполнение данного проекта может способствовать развитию логического и творческого мышления у учащихся.

Задача проекта «Арканоид»: Смысл игры заключается в том, что игроку необходимо перемещать платформу-ракетку влево-вправо и отбивать шарик, который постоянно летит вниз. Если игрок отбил шарик, то он должен полететь в противоположную сторону и разрушать кирпичи, которые находятся сверху. Если шарик упал ниже платформы, то игра начинается заново. Каждый раз, когда игрок отбивает шарик у него повышается скорость.

Выполнение данного проекта учениками может преследовать несколько целей:

– Обучающая: ближе познакомятся с понятием переменных, условных операторов, познакомятся с некоторыми объектно-ориентированными средствами, клонированием объектов;
– Развивающая: выполнение данного проекта может способствовать развитию логического и творческого мышления у учащихся.

Задача проекта «Догонялки»: разработать игру догонялки, по ходу которой игроку предстоит догонять другого персонажа. По правилам догоняемый может вырваться из рук догоняющего, так что его предстоит догнать несколько раз (сколько конкретно раз необходимо догнать решает программист). Так как догоняемый вырывается, то ему нужно отдохнуть, т.е. ничего не делать. Именно в это время игрок и может его догнать. После того как игрок окончательно поймает догоняемого, они оба появляются в центре, и игра считается оконченной.

Выполнение данного проекта учениками может преследовать несколько целей:

– Обучающая: ближе познакомятся с понятием переменных, условных операторов, познакомятся с некоторыми объектно-ориентированными средствами, клонированием объектов;

– Развивающая: выполнение данного проекта может способствовать развитию логического и творческого мышления у учащихся.

Приведенные выше задания желательно давать после того, как учащиеся будут знать такие понятия как переменная, условный оператор, ориентируются в прямоугольной системе координат и понимают работу сенсоров. В случае, если в классе есть, как сильные, так и слабые ученики, то сильным можно к этим задачам придумать дополнительные критерии оценивания. Например, чтобы в проекте «Полет» у учеников была также преграда сверху, а надвигающиеся преграды были разнообразными.

Приведем памятку при работе в Scratch, которая содержит приемы, универсальные при создании всех проектов. В большей степени она предназначена для учителей, однако её можно показать и ученикам, но строго после того, как они попробуют реализовать данные действия самостоятельно.

Создание переменной для подсчета полученных очков

1. Перейти во вкладку переменные;
2. Создать переменную с необходимым именем;
3. Подставить в код блок «Изменить <Переменная> на n».

Переход спрайта в случайную позицию

1. Выбрать один из трех циклов на вкладке «Управление»;
2. На вкладке «Движение» выбрать «Перейти на <Случайное положение>» и вставить в блок цикла;
3. Вставить блок «Ждать» на вкладке «Управление», чтобы переход спрайта не был мгновенным.

Перемещение вправо, влево, вверх, вниз

1. Выбрать блок «Когда клавиша <стрелка> нажата» из вкладки «События»
2. Присоединить к блоку события блоки «Изменить x на n» или «Изменить y на n» из вкладки «Движение» в зависимости от того, на какую стрелку требуется нажать для перемещения.

Создание анимации

1. Добавить спрайт и убедится, что у него есть упорядоченные для анимации костюмы;
2. Выбрать один из трех циклов на вкладке «Управление»;
3. Во вкладке «Внешний вид» вставить в цикл блок «Следующий костюм»
4. Вставить блок «Ждать» на вкладке «Управление» для более плавной анимации (рекомендуем ожидание в 0,3 секунды).

Создание процедуры

1. Выбрать вкладку другие блоки;
2. Создать блок, определив необходимые типы аргументов (переменная, условие, строка);
3. Определить созданный блок, например, определить блок по движению персонажа, вместо 4 отдельных событий, использовать одно, но с применением четырех условий;
4. Вызвать данный блок, подставив требуемую переменную, условие или строку (блок доступен лишь спрайту, в котором он был определен).

2.3 Апробация разработанных материалов

Во время педагогической практики разработанные учебные проекты были показаны учителю высшей категории лицея 110 Неуйминой Наталье Германовне на что было получено экспертное заключение. Отзыв находится в приложении 1.

Заключение

Подводя итоги проведенного исследования можно сказать, что были достигнуты следующие результаты:

1. Были выявлены особенности развития в младшем подростковом возрасте:
   - в данном возрасте ученики активно ищут себя
   - для учеников данного возраста общение становится одним из видов деятельности
   - в связи с проблемами с контролем учеников за своими действиями, требуются простые в освоении средства обучения
2. Во время сравнения сред программирования были:
   - Проанализированы популярные исполнители и среды программирования
   - Выявлены их плюсы и минусы
   - Изучена подробно среда программирования Scratch

Разработаны:

- учебные проекты
- методические рекомендации к их применению в обучении
- памятка по реализации основных приемов при создании проектов в Scratch

В ходе исследования мы выяснили, что среда Scratch предоставляет возможность создания качественных проектов, предлагая пользователям богатый инструментарий и при этом, являясь средой, легкой для освоения. Применение данной среды программирования в 5-6 классах позволит ученикам развивать логическое и алгоритмическое мышление, познакомиться с многими понятиями программирования, облегчит знакомство с более профессиональными средами программирования в дальнейшем.

Библиографический список

1. Борович П.С., Бутко Е.Ю. Учебное пособие "Среда программирования Scratch".
2. Голиков Д, Голиков А. Книга юных программистов на Scratch. Los Gatos: Smashwords, 2013.
3. Д.Б. Эльконин Избранные психологические труды. М.: "Педагогика", 1989.
4. Дидактика и информатика // Энциклопедия учителя информатики URL: http://inf.1september.ru/2007/11/08.htm (дата обращения: 21.04.2018).
5. И.В. Дубровина, А.М. Прихожан, В.В. Зацепин Возрастная и педагогическая психология: Хрестоматия: Учеб. пособие для студ.высш.учеб.заведений. М.: Академия, 1999. 320 с.
6. Исполнители // kpolyakov.spb.ru URL: goo.gl/pGSL6Z (дата обращения: 5.04.2018).
7. Исполнители алгоритмов // sgls.ru URL: https://goo.gl/spfwwB (дата обращения: 8.04.2018).
8. КуМир // Система программирования КуМир URL: https://www.niisi.ru/kumir/index.htm (дата обращения: 16.04.2019).
9. КуМир // Wikipedia URL: https://ru.wikipedia.org/wiki/КуМир (дата обращения: 17.04.2019).
10. Логическое мышление - развитие логики // 4BRAIN URL: https://4brain.ru/logika/#1 (дата обращения: 20.04.18).
11. Лого (язык программирования) // Wikipedia URL: https://clck.ru/9oDHp (дата обращения: 16.04.2019).
12. ЛогоМиры 3.0. Интегрированная творческая среда // Институт новых технологий URL: https://goo-gl.ru/5c8O (дата обращения: 15.04.2019).
13. Методика преподавания темы "Программирование в среде Scratch" учащимся начальной школы // Age Pedagogy URL: http://www.agepedagog.ru/grepn-362.html (дата обращения: 9.04.2018).
14. Мухина В.С Возрастная психология. Феноменология развития. 10 изд. М.: «Академия», 2006.
15. Мышление // Национальная психологическая энциклопедия URL: https://vocabulary.ru/termin/myshlenie.html (дата обращения: 13.04.2018).
16. Пономарев М.В., Рожина И.В. ЭЛЕМЕНТЫ МЕТОДИКИ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ В СРЕДЕ SCRATCH УЧАЩИХСЯ // АКТУАЛЬНЫЕ ВОПРОСЫ ПРЕПОДАВАНИЯ МАТЕМАТИКИ, ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ. Екатеринбург: 2019.

17. Приемы и методы мышления // http://textb.net URL: http://textb.net/105/68.html (дата обращения: 20.04.2018).
18. Свойства алгоритма // http://shkolo.ru URL: http://shkolo.ru/svoystva-algoritma/ (дата обращения: 20.04.2018).
19. Скретч (язык программирования) // Википедия URL: goo.gl/J7kcNc (дата обращения: 8.04.2018).
20. Стили мышления и обучение программированию студентов педагогического вуза // Информационные технологии в образовании URL: http://ito.edu.ru/2006/Moscow/I/1/I-1-6371.html (дата обращения: 21.04.2018).
21. Тетеревкова А.В. Формирование мотивации к творческой деятельности у детей младшего подросткового возраста на занятиях в кружке декоративно-прикладного творчества // Психология сегодня: материалы XI региональной научно-практической конференции студентов и аспирантов. Екатеринбург: 2009.
22. Уфимцева П.Е., Рожина И.В. Обучение программированию младших школьников в системе дополнительного образования с использованием среды разработки SCRATCH //Наука и перспективы. 2018. № 1. С. 29-35.
23. ФГОС основного общего образования (5 - 9 кл.) // ФГОС URL: https://fgos.ru (дата обращения: 20.05.2019).
24. Шаповаленко И.В. Возрастная психология (психология развития и возрастная психология). М.: «Гардарики», 2005.
25. Bob Johnstone Never Mind the Laptops: Kids, Computers, and the Transformation of Learning. Lincoln: iUniverse, 2003.
26. Logo Tree Project // elica.net URL: https://goo-gl.ru/5c8N (дата обращения: 15.04.2019).
27. Project Types // en.scratch-wiki.info URL: https://en.scratch-wiki.info/wiki/Project_Types (дата обращения: 9.04.2018).
28. Scratch // Progopedia.ru URL: http://progopedia.ru/language/scratch/ (дата обращения: 8.04.2018).

Приложение

Приложение 1.
Отзыв на разработку «Система учебных проектов для среды программирования Scrath»

Актуальность. Формирование алгоритмического мышления в младшем и среднем школьном возрасте требует иных подходов чем обще принятое обучение программированию. Среда Scratch позволяет использовать игровые технологии в процессе обучения, что повышает мотивацию школьников в освоении программирования. На официальном сайте Scratch представлены материалы для базового освоения языка и среды программирования, существует ряд методической литературы по освоению среды и базовых алгоритмических конструкций. Например, следует сказать о «Книге юных программистов» (автор Голиков Д.), учебном пособии «Среда программирования Scratch» (Борович П.С. и Бутко Е.Ю.), в которых представлен ряд лабораторных работ, обучающих программировать в среде Scratch. Данные работы ориентированы на учеников 3-5 классов. Кроме того, можно найти много видеороликов с похожим содержанием на площадке YouTube.

Однако, все они ориентированы на репродуктивную деятельность. В тоже время ФГОС требует развития деятельностного метода обучения, выработки у учащихся алгоритмической культуры, создание условий для активного участия в исследовательской деятельности. Данная разработка пытается привнести частично-поисковую и исследовательскую деятельность в процесс обучения, что является, несомненно, актуальным в современных условиях.

Особенности и общее описание разработки. В рамках исследовательской работы «Обучение программированию с использованием визуальной событийно-ориентированной среды Scratch в курсе информатики и ИКТ» были реализованы пять проектов в среде программирования Scratch и разработано их методическое сопровождение по следующим темам:

1. **Питомец** – аналог игры Тамагочи или «Кот Том». Игроку требуется кормить, поить и укладывать спать питомца. Для реализации проекта использовались основные алгоритмические конструкции – циклы, условные операторы, переменные и некоторые особенности самих блоков Scratch.
2. **Арканоид** – клон одноименной игры. Игроку необходимо при помощи мышки управлять платформой и отбивать мяч, который должен сбить кирпичи, летящие сверху. Для реализации проекта использовались основные алгоритмические структуры и работа с сенсорами, например, управление платформой при помощи компьютерной мыши.
3. **Полет** – клон игры FlappyBird. Игроку необходимо лететь сквозь город и стараться не прикасаться к зданиям.
4. **Прыжки** – в данной игре необходимо перепрыгивать через препятствия.

5. **Догонялки** – в данной игре необходимо определенное количество раз догнать предмет, который каждый раз выскальзывает из рук игрока и оказывается в случайном месте на поле. В некоторой степени проект можно считать, как проект повышенной сложности, т.к. кроме стандартных алгоритмических конструкций используются также методы.

В методических рекомендация, разработанных авторами данной разработки, для каждого проекта приводится описание создания проекта, как «идеального» образца, тексты постановки заданий и преследуемые цели обучения для каждого проекта.

В процессе выполнения предлагаемых проектов учениками преследуется несколько педагогических целей:

Обучающая: освоение понятия переменных, условных операторов, циклических конструкций, знакомство с типовыми алгоритмами, некоторыми объектно-ориентированными средствами.

Воспитательная: тематика проектов служит воспитанию гуманности в учениках.

Развивающая: выполнение проектов способствует развитию логического и творческого мышления у учащихся.

В ходе апробации реализации данных проектов с учащимися 5 классов было установлено, что:

- групповая форма деятельности при выполнении проекта дает более высокие результаты;
- использование игровых моментов повышает мотивацию;
- необходимо дополнительно разработать приемы, повышающие концентрацию внимания учащихся в ходе выполнения проектов.

Таким образом, исследовательская работа по теме «Обучение программированию с использованием визуальной событийно-ориентированной среды Scratch в курсе информатики и ИКТ» дает возможность развития алгоритмического мышления учащихся младшего и среднего звена в соответствии с требованиями ФГОС и методические рекомендации, представленные в ней, могут быть использованы в учебном процессе.

Неуймина Наталья Германановна,
учитель информатики высшей категории
МАОУ лицея №110 им. Л.К. Гришиной,
руководитель педагогической практики
студентов ИМФИиТ УрГПУ